



Karpatica Travel

Proiect Inginerie Software 2021-2022

Membrii echipei:

Cobzaru Mihai, 10LF291

Leca Marian Rafael, 10LF292

Neacsu Alexandru Cristian, 10LF293

Petre Mihai Bogdan, 10LF293 - Team Leader

Coordonator:

Maria Nutu

Cuprins

1. Funcționalitățile aplicației.....	3
2. Etapele proiectării aplicației.....	3
3. Prezentarea tehnologiilor folosite	
3.1 Server BD.....	4
3.2 Limbaje de programare.....	4
3.3 ORM.....	4
3.4 Creare diagrame.....	5
3.5 Versionare	5
3.6 Management task-uri.....	5
3.7 Creare GUI	6
3.8 Modul de comunicare client-server.....	6
3.9 Testarea aplicației.....	6
3.10 Descrierea unor scenarii de utilizare.....	7
4. Diagrame.....	9
4.1 Diagrama cazurilor de utilizare.....	9
4.2 Diagrama bazei de date.....	10
4.3 Diagrama de clase.....	11
4.4 Diagrama de stare.....	14
4.5 Diagrama de colaborare.....	14
4.6 Diagrama de activități.....	15
4.7 Diagrama de interacțiuni.....	16
4.8 Diagrama de componente.....	17
4.9 Diagrama de pachete.....	18
4.10 Diagrama de desfășurare.....	19

1. Funcționalitățile aplicației

Prin acest proiect ne propunem să creăm o agenție de turism online. Clienții au posibilitatea de a-și planifica următoarea vacanță.

Un server va fi folosit pentru a stoca toate detaliile despre cazarea pe care agentia noastra le poate oferi clienților, respectiv informațiile referitoare la rezervările efectuate.

Un API este utilizat pentru a prelua date de pe server și a popula pagina web.

Utilizatorul poate să vizualizeze o listă de cazări posibile filtrate în funcție de mai multe criterii precum preț, distanța față de centrul orașului, o anumită țară sau oraș. Odată ce a găsit o cazare pe placul său, utilizatorul poate să facă o rezervare dacă cazarea respectivă are locuri destule.

Totodată, user-ul poate să-și verifice rezervările făcute și le poate anula până la data cazării. Pentru a-și exprima satisfacția în legătură cu cazarea rezervată, utilizatorul poate să lase o recenzie, care va fi ulterior văzută pe pagina cazării.

În plus, există posibilitatea de administrare a site-ului astfel încât adminul poate să adauge sau să șteargă cazări în funcție de nevoie.

2. Etapele proiectării aplicației

Faza inițială a proiectării aplicației a constat în întocmirea diagramei bazei de date și stabilirea modului în care componentele aplicației interacționează între ele.

Pasul următor a fost să implementăm backend-ul aplicației pe baza cerințelor bazei de date. Odată ce am implementat partea API-ul care permite înregistrarea și logarea utilizatorilor, în paralel am început dezvoltarea părții de frontend a aplicației.

3. Prezentarea tehnologiilor folosite

3.1 Server BD

Pentru baza de date am ales sa folosim Microsoft SQL Server 2019, cu un server hostat pe un serviciu de hosting. SQL Server este un sistem sistem de gestionare a bazelor de date relational, care permite stocarea și interogarea datelor pe server.

3.2 Limbaje de programare

Limbajele folosite pentru dezvoltarea aplicației sunt C# pentru partea de backend, și typescript, în combinatie cu HTML pentru partea de frontend. API-ul folosit este implementat în framework-ul ASP .NET Core, în timp ce frontend ul este realizat cu ajutorul framework-ului de javascript Angular.

3.3 ORM

Legatura dintre API-ul aplicației și baza de date este facilitată de către folosirea ORM-ului Entity Framework Core. EF Core este un framework pentru aplicații de tip .NET care permite dezvoltatorilor să lucreze cu clase specifice domeniului aplicației, acest ORM realizand maparea datelor aplicației în baza de date fără să aibă nevoie de intervenția dezvoltatorilor.

3.4 Creare diagrame

Diagramele au fost create în mod colaborativ cu toți membrii echipei în cadrul platformei draw.io, aceste fiind ulterior salvate și încărcate pe repository-ul de github al proiectului.

3.5 Versionare

Pentru managementul versionării proiectului am creat un repository pe [github](#). Fiecare membru al echipei are un branch separat în funcție de ce funcționalitate a aplicației implementează/modifică, iar în final pentru a putea aduce schimbările făcute pe branch-ul de main trebuie făcut un pull request care trebuie aprobat de restul membrilor echipei.

Link către repo: <https://github.com/PetreBogdan55/KarpaticaTravel>

Cele 10 branch-uri utilizate sunt: frontend-MC, frontend-LMR, frontend-NAC, frontend-PMB, API-MC, API-LMR, API-NAC, API-PMB, dev, main.

3.6 Management task-uri

Managementul de task-uri a fost realizat prin platforma [trello](#), unde fiecare membru al echipei a avut mai multe task-uri atribuite pe măsura desfășurării proiectului.

Fiecare task a fost împărțit în funcție de partea aplicației pe care o viza (API/Frontend/Baza de date) și de nivelul de prioritate al acestuia. De asemenea, task-urile au avut și o estimare a gradului de dificultate și un checklist cu obiective care trebuiesc îndeplinite.

Link către trello: <https://trello.com/b/b0HL6pM7/karpaticatravel>

3.7 Creare GUI

Interfata grafica a fost implementată în framework ul Angular, care permite împărțirea aplicației în anumite componente de pot fi reutilizate.

Inițial, am creat pagina de înregistrare și de login a utilizatorului, după care am dezvoltat paginile de home si start, care servesc drept o mica introducere in aplicatia noastra.

Ulterior, a fost implementată pagina care returnează rezultatele căutării de cazari in funcție de anumite criterii, împreuna cu paginile care afișează rezervările, respectiv recenziile utilizatorului.

În final, am adaugat și pagina de administrare a site-ului.

3.8 Modul de comunicare client-server

Modulul de comunicare dintre client și server (API) a fost creat cu ajutorul framework-ului .NET. Acesta suporta operații CRUD (Create, Read, Update, Delete) pentru următoarele tipuri de entități: User, Location, Booking, Review, City, Country, Activity.

Pentru a interactiona cu API-ul aplicației, în componenta de frontend au fost create niște servicii care îndeplinesc rolul de intermediar între frontend și API, acestea apelând endpoint-urile care se găsesc pe backend.

Fiecare request făcut de utilizator este preluat de către serviciul aferent în frontend, care trimite informațiile mai departe către API, acesta din urma comunicand cu baza de date pentru a efectua operația dorită.

3.9 Testarea aplicației

Înainte de trimiterea oricărei modificări spre repository, fiecare functionalitate a trecut prin procesul de consultare cu membrii echipei și de testare manuala in cadrul aplicatiei.

Eventualele bug-uri găsite au fost rezolvate și încă o data testate manual pentru a asigura funcționarea corectă.

3.10 Descrierea unor scenarii de utilizare

Înregistrarea și autentificarea unui utilizator *Register*

The screenshot shows the 'Register' form within the KarpaticaTravel 1.0.0 application. The form is titled 'Register' and contains the following fields and elements:

- Username ***: A text input field with a red error message below it: '*Please enter username'.
- Email ***: A text input field with a red error message below it: '*Please enter your email'.
- Phone ***: A text input field with a red error message below it: '*Please enter your email'.
- Password ***: A text input field with a red error message below it: '*Please enter your password'.
- Non sono un robot**: A checkbox with a red error message below it: '*Please enter your email'.
- reCAPTCHA**: A reCAPTCHA widget with a red error message below it: '*Please enter your email'.
- Register**: A button to submit the form.
- Go to login**: A button to navigate to the login page.

Login

The screenshot shows the 'Login' form within the KarpaticaTravel 1.0.0 application. The form is titled 'Login' and contains the following fields and elements:

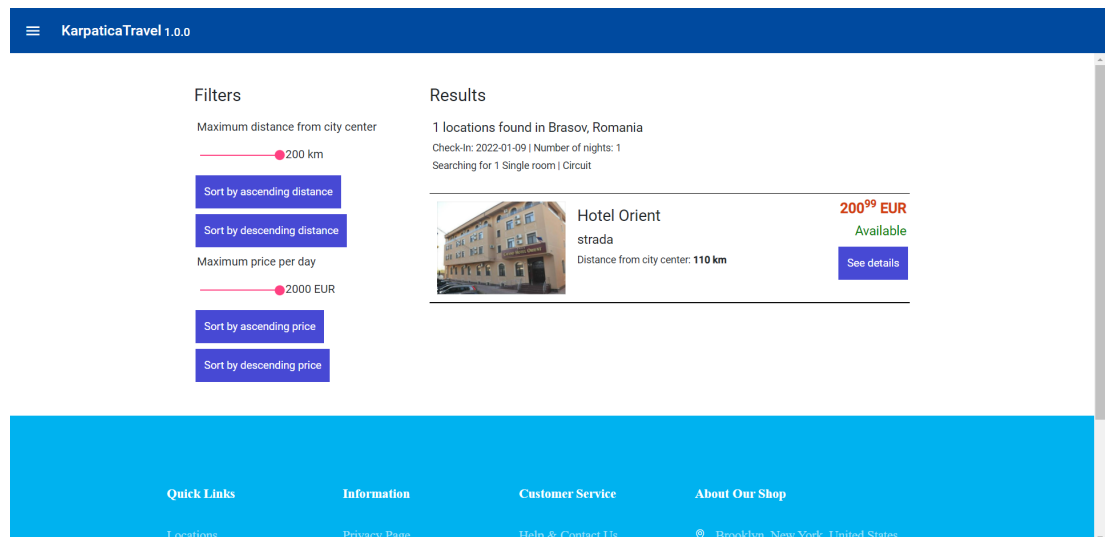
- Email ***: A text input field with a red error message below it: '*Please enter your email'.
- Password ***: A text input field with a red error message below it: '*Please enter your password'.
- Sign-in**: A button to submit the form.
- Go to register**: A button to navigate to the register page.

- Utilizatorul, dacă nu are un cont existent, poate alege opțiunea de înregistrare pentru a fi introdus în baza de date a aplicației.
- În cazul a făcut înregistrarea, user-ul se poate loga cu emailul si parola sa.

Căutarea unei cazări

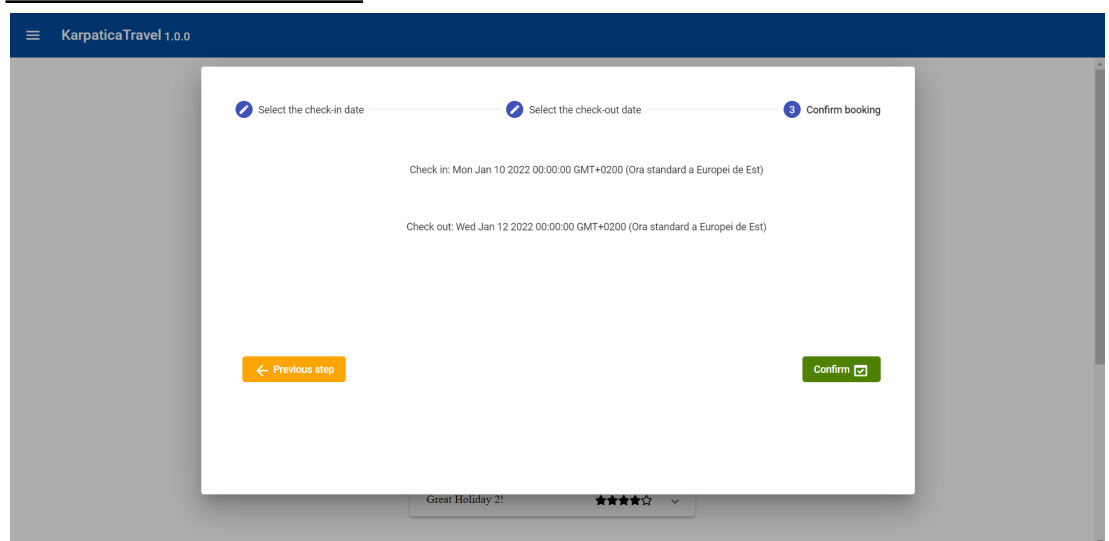
The screenshot shows the 'Search for a location...' form within the KarpaticaTravel 1.0.0 application. The form is titled 'Search for a location...' and contains the following fields and elements:

- Resort**: A dropdown menu.
- Romania, Brasov**: A text input field.
- 2022-01-10, 1 nights**: A date and duration input field.
- Choose your room**: A dropdown menu.
- Search**: A button to submit the form.



- Pentru căutarea unei cazări, user-ul poate selecta anumite criterii în bara de search, precum tipul cazarii, orașul dorit și data check-in-ului.
- De asemenea, utilizatorul poate sa caute cazari in funcție de activitățile aferente sau de orașul dorit.

Realizarea unei rezervari

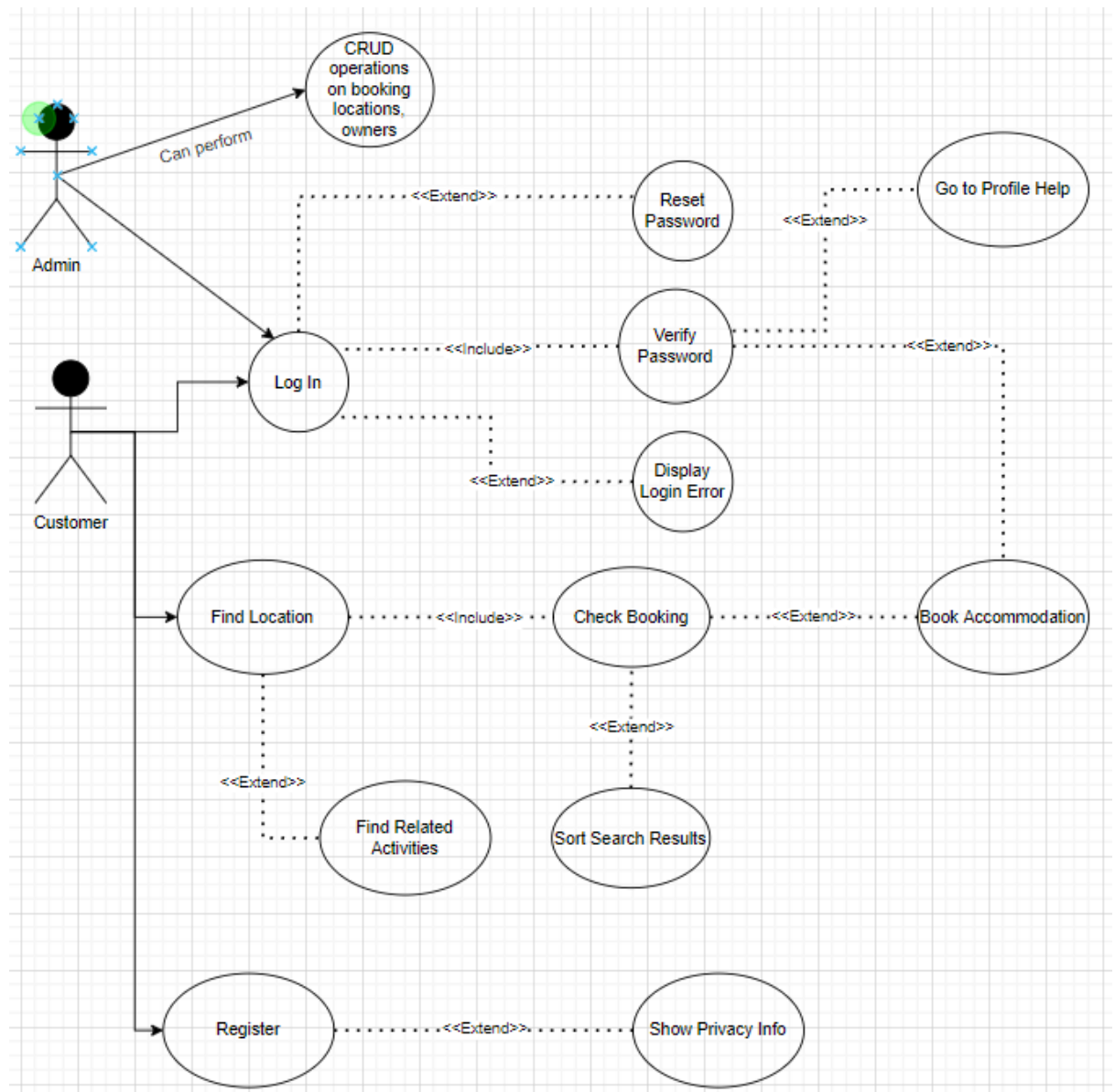


- Odată găsită cazarea dorită, se poate face o rezervare apasand pe butonul de book location, dacă aceasta este valabilă.
- Dacă butonul de rezervare a cazarii este apăsat, user-ul va fi nevoit sa selecteze datele pentru check-in și check-out, iar la final sa confirme rezervarea.

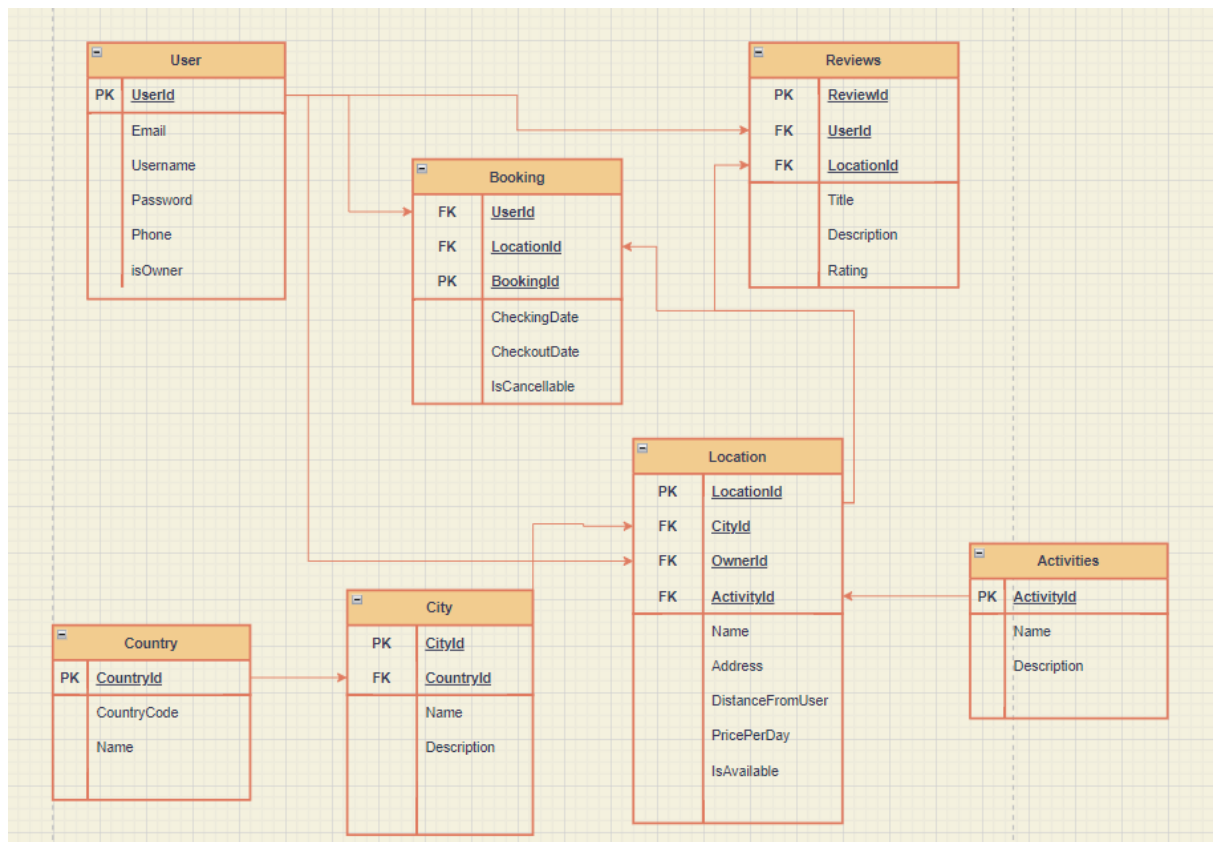
4. Diagrame

Mai jos am atașat cate o poza a fiecărei diagrame:

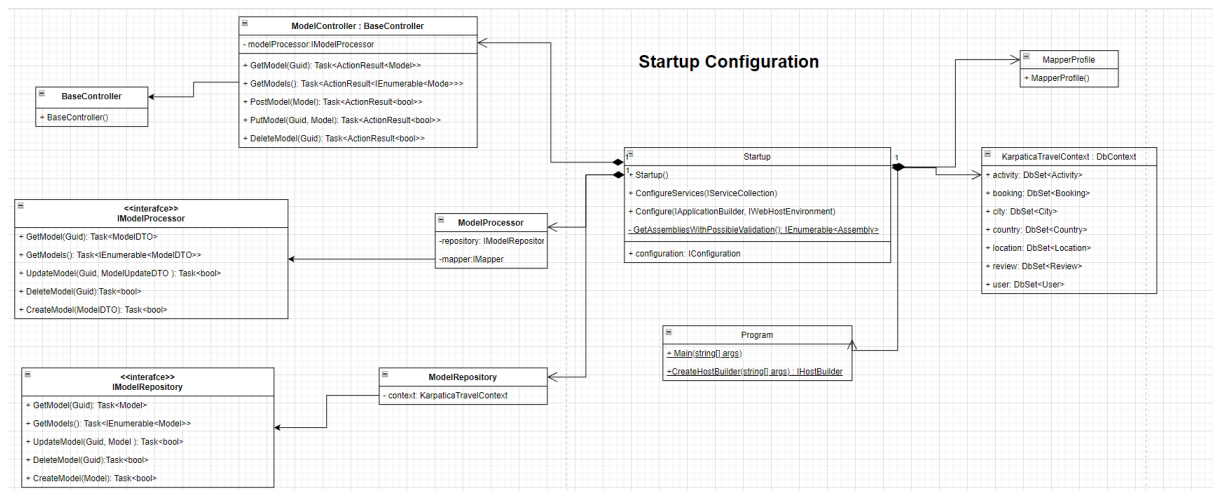
4.1 Diagrama cazurilor de utilizare



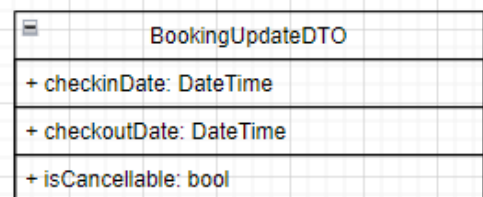
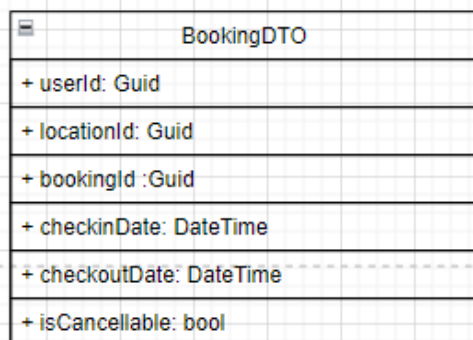
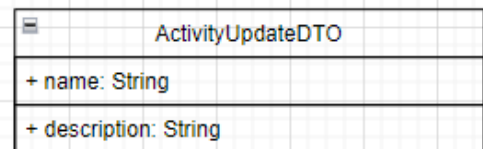
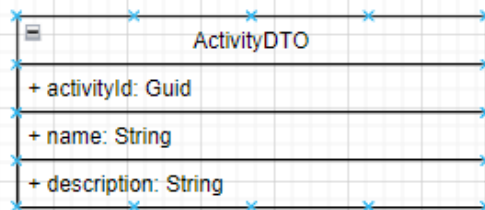
4.2 Diagrama bazei de date



4.3 Diagrama de clase



DTO Entities



CityDTO
+ cityId: Guid
+ countryId: Guid
+ name: String
+ description: String

CityUpdateDTO
+ name: String
+ description: String

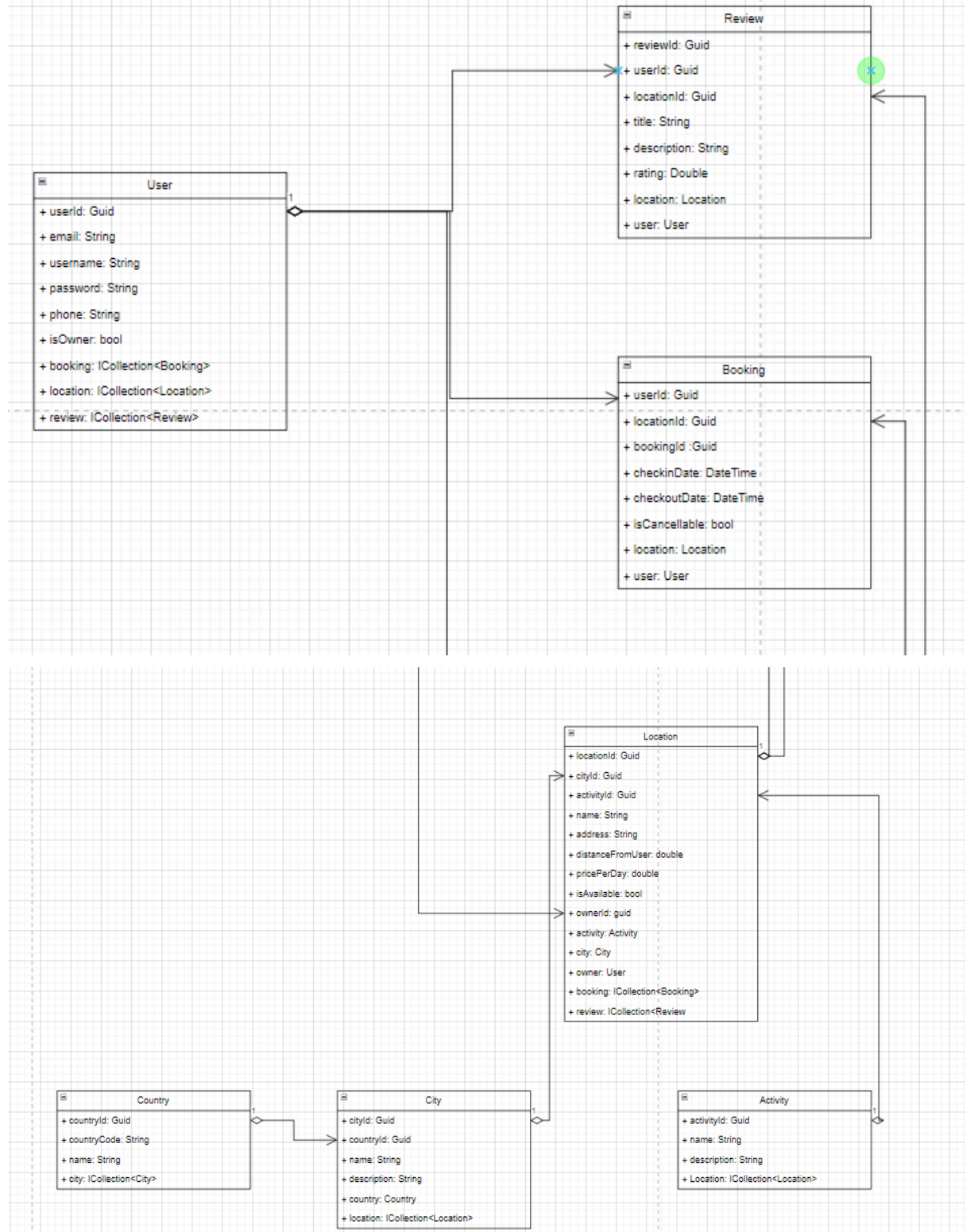
CountryDTO
+ countryId: Guid
+ countryCode: String
+ name: String

CountryUpdateDTO
+ countryCode: String
+ name: String

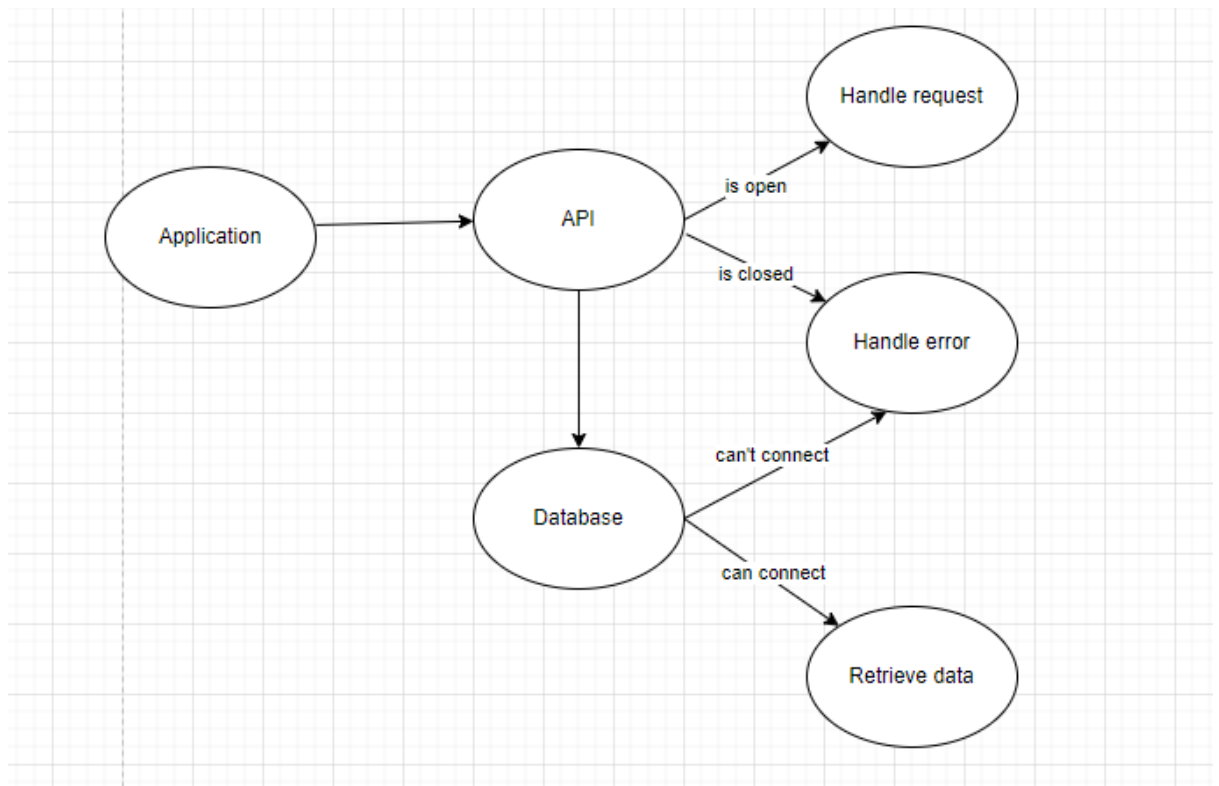
LocationDTO
+ locationId: Guid
+ cityId: Guid
+ activityId: Guid
+ name: String
+ address: String
+ distanceFromUser: double
+ pricePerDay: double
+ isAvailable: bool
+ ownerId: guid

LocationUpdateDTO
+ name: String
+ address: String
+ distanceFromUser: double
+ pricePerDay: double
+ isAvailable: bool

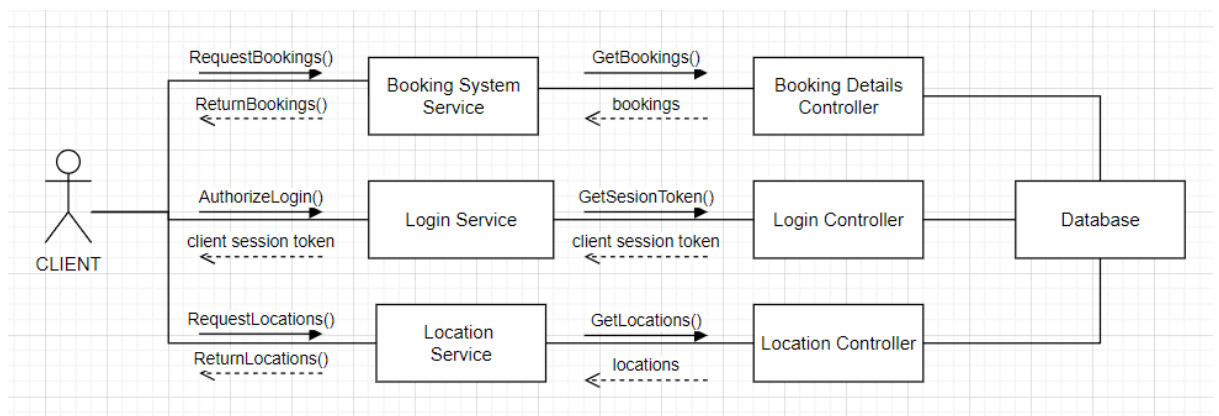
Entity Classes



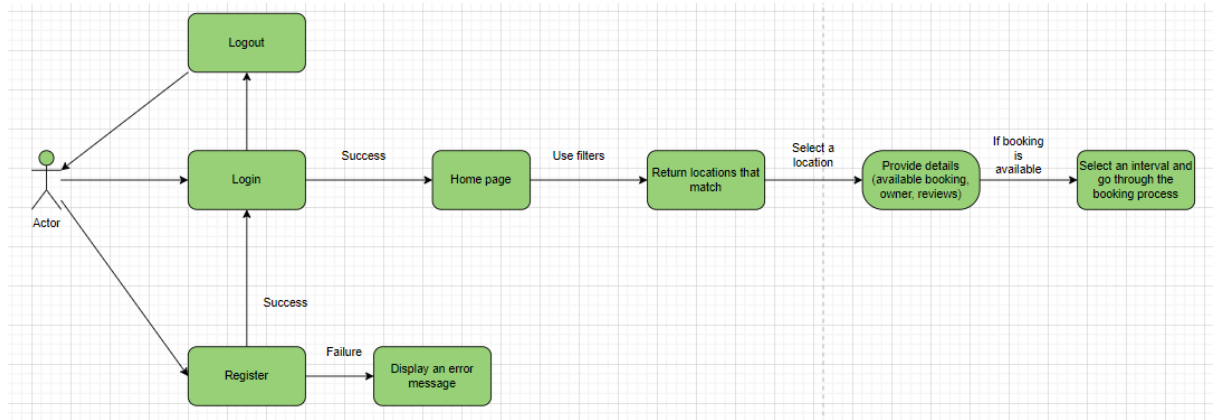
4.4 Diagrama de stare



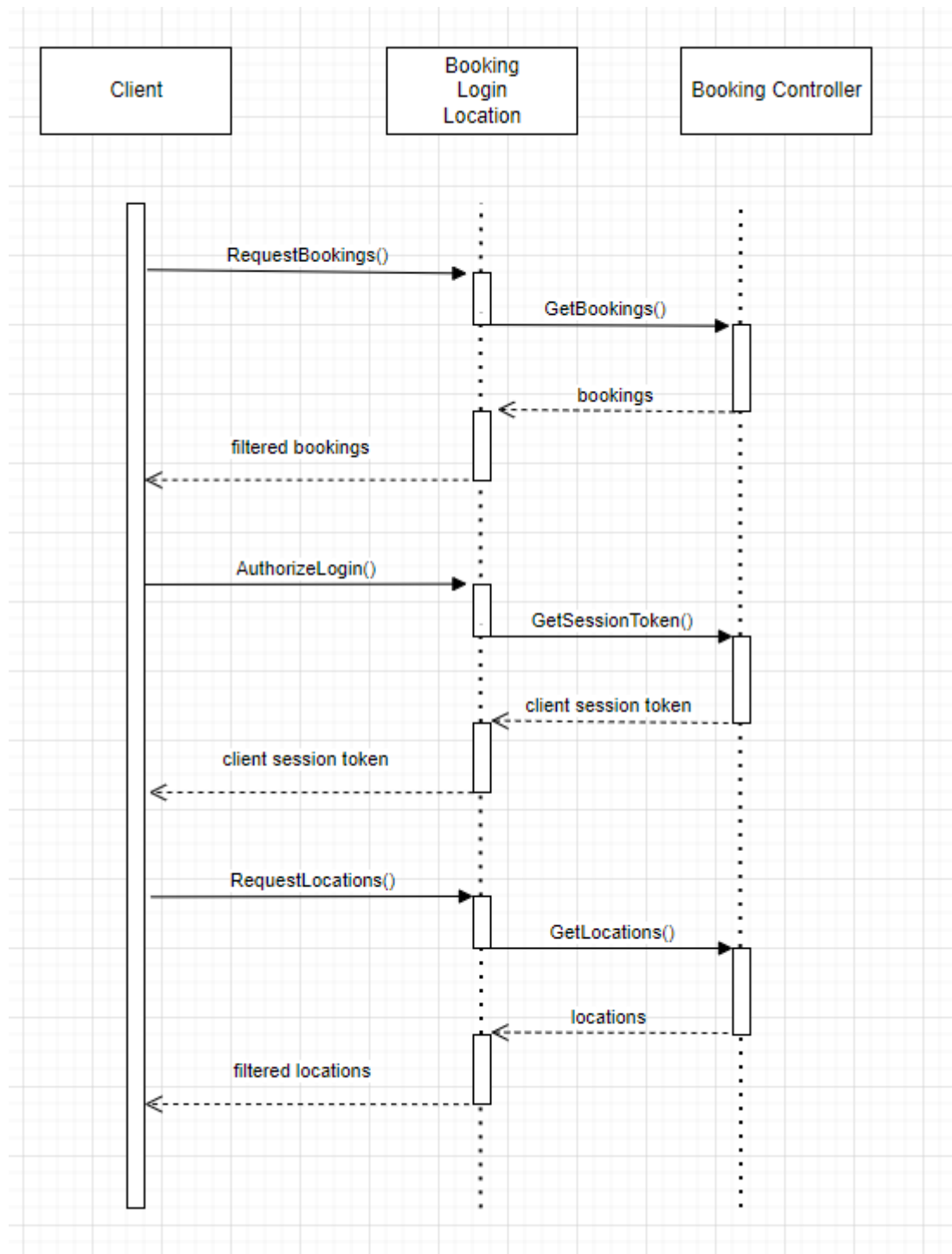
4.5 Diagrama de colaborare



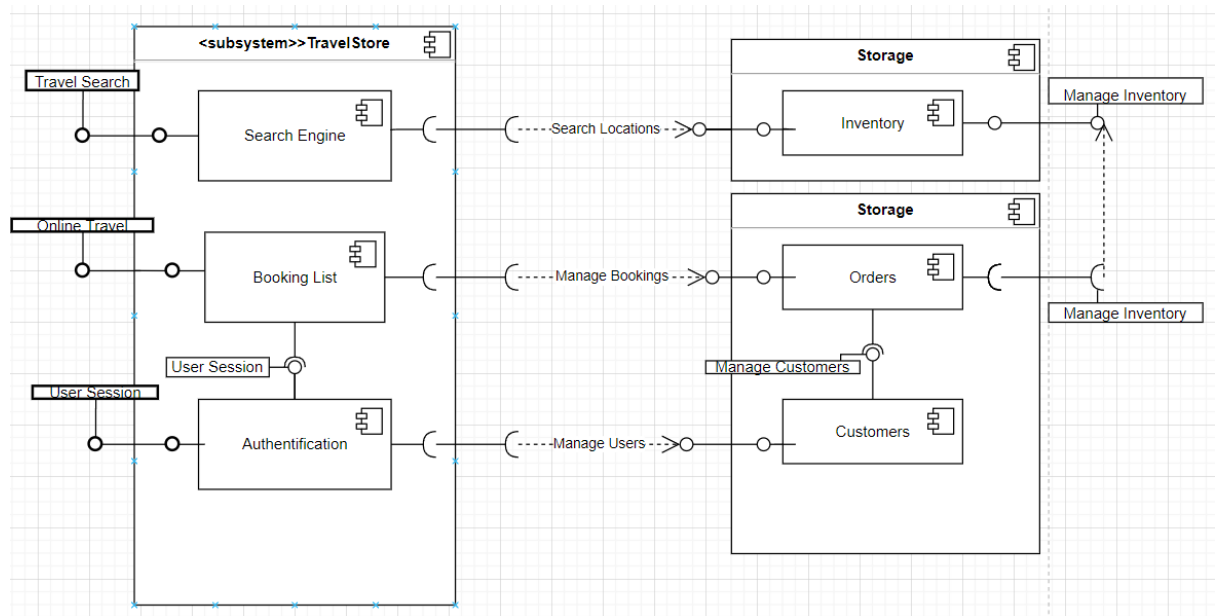
4.6 Diagrama de activități



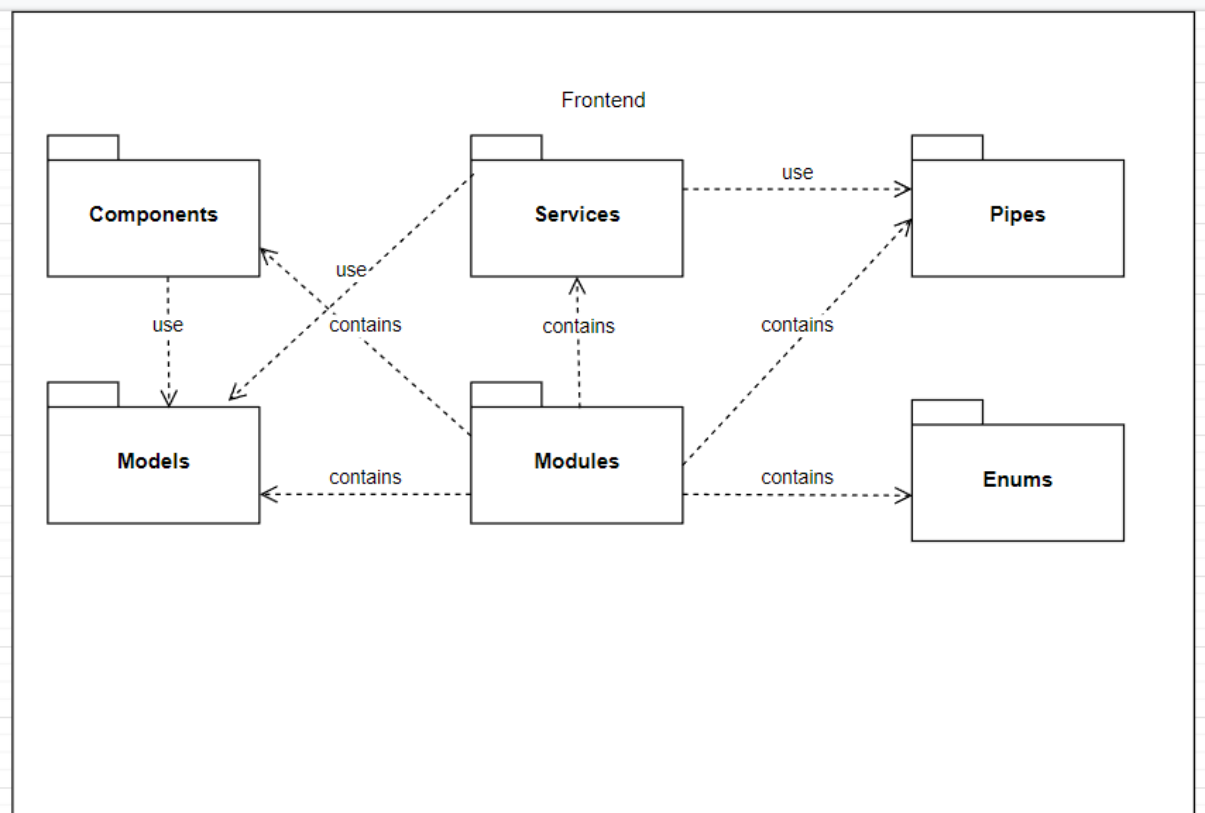
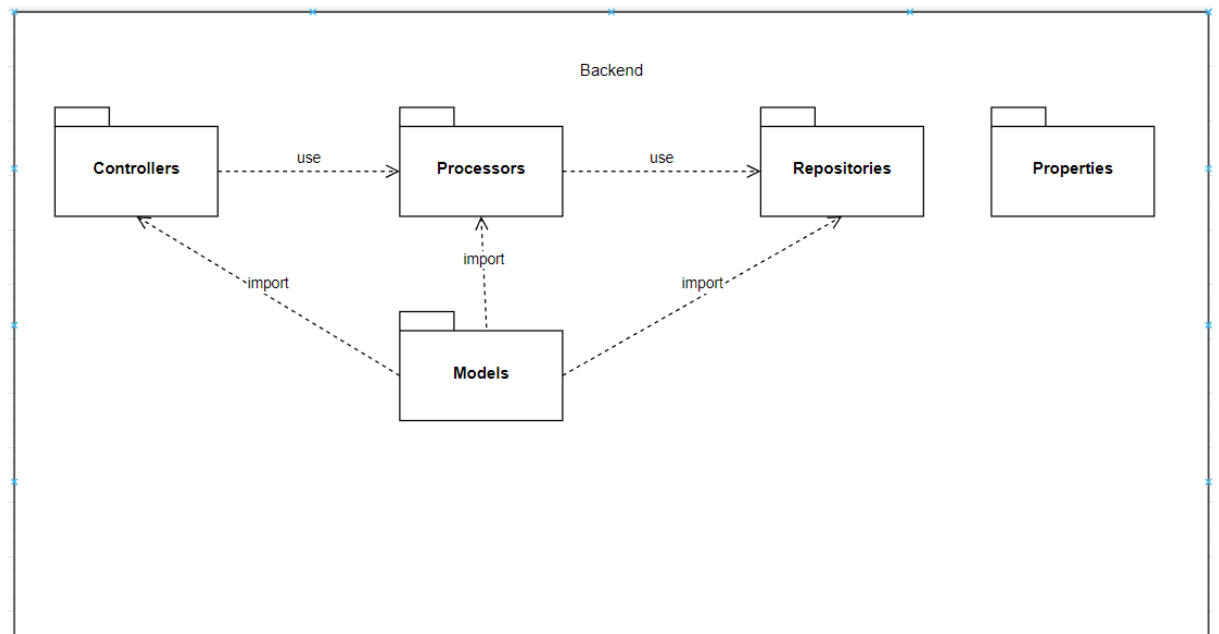
4.7 Diagrama de interacțiuni



4.8 Diagrama de componente



4.9 Diagrama de pachete



4.10 Diagrama de desfășurare

